LKS SMK
Tingkat Nasional Ke-XXVII
Tahun 2019

Soal

LKS SMK

BIDANG LOMBA

IT-Software Solution for Business

SMK Bisa !
Indonesia Bisa !

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
DIREKTORAT JENDERAL PENDIDIKAN MENENGAH
DIREKTORAT PEMBINAAN SEKOLAH MENENGAH KEJURUAN
Kompleks Kemdiknas Gedung E Lantai 12-13
Jalan Jenderal Sudirman Senayan Jakarta 10270
Telepon (021) 5725477 (hunting), 5725466-69, 5725471-75
Fax. 5725467, 5725469, 5725049
Site: www.ditpsmk.net

PROJECT OVERVIEW

In this Test Project, you are required to develop an application following the requirements given on the Test Project. There will be 4 project resources given to you, they are:

1. **ERD and Data Dictionary for Test Project**
   This will be used to guide you to create the database files. Ensure that all entities created on the database are related with the given ERD, following the relationship and also the criteria in Data Dictionary for each entity.

2. **Navigation Diagram for the Application**
   Use the Navigation Diagram to develop the form interaction inside the application. You are free to add new interaction between forms, without neglecting the defined interaction in this diagram.

3. **Example Design (Wireframe) of desired Information System**
   This file is used to give you design guidelines of all required forms. Please note that your form designs are not limited to these examples.

4. **Data Files (*if any*)**
   The data files will be used to support your test and development of the information system of each module. Please use this data files in the development. You can use other resources outside the given data files, only if it is instructed in the module.

GENERAL GUIDELINES

1. **Components**

   Ensure that all input fields use appropriate components. Assume that the testing will be carried out by new user who never use this or any similar systems before.

2. **Database Connection**

   Use SQL Server as your database. Use below setting for the connection:

Database Name	**PC_XX (XX is PC number)**
Server	**.\SQLEXPRESS**
Username	**Windows authentication**
Password	

3. **Answer Submission**

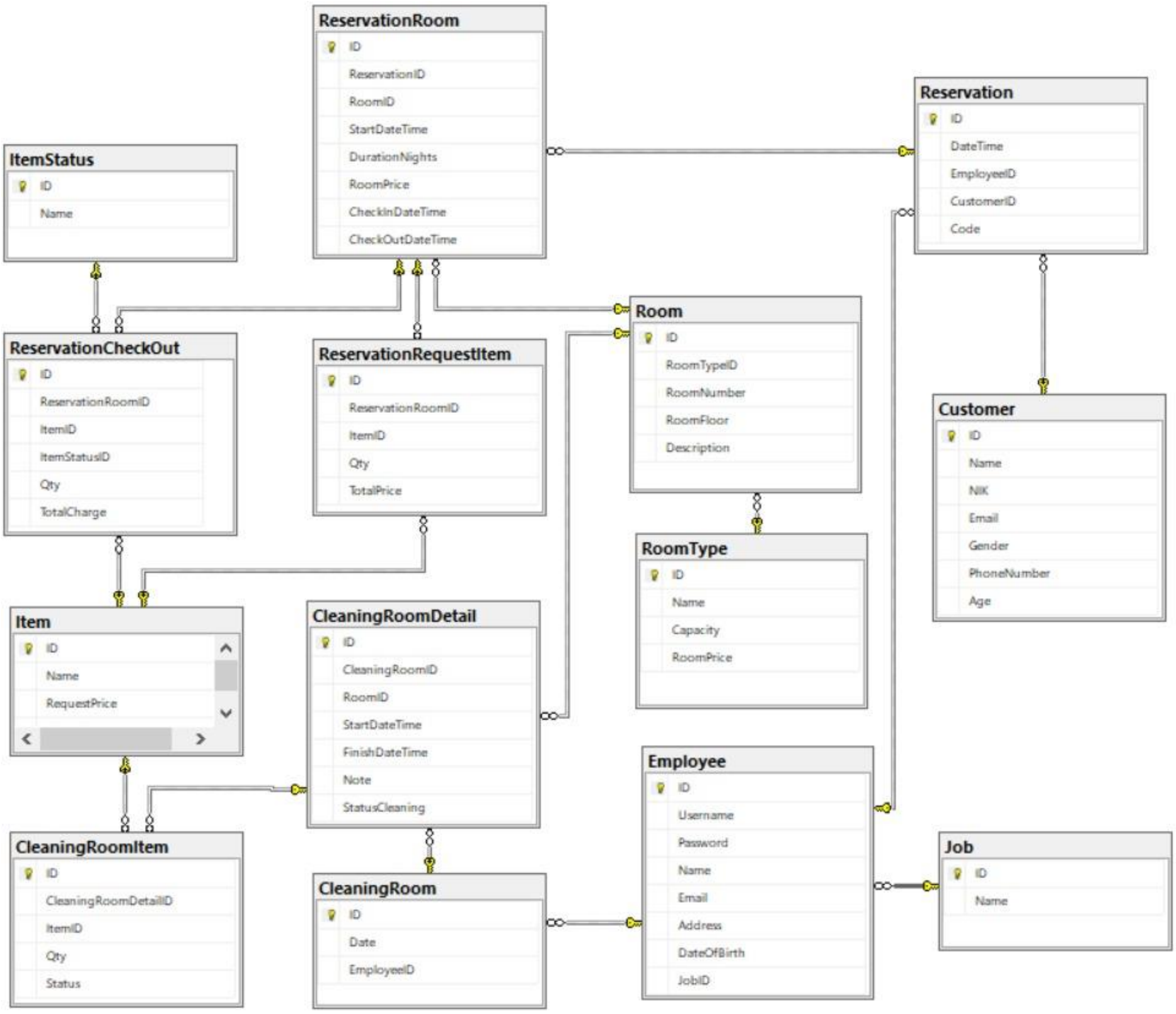
   At the end of each session, submit your answer inside a folder with format: **PC_XX_MODUL_YY** (XX is PC number and YY is session). Your submission must at least contain:

   - ✓ SQL Script to recreate your database in testing machine. The script must contain table creation, relationship definition, constraints definition, insert data, and along with other, such as stored procedures, cursors, or views if any. Rename the file to "PC_XX.sql", where XX is PC number.
   - ✓ All source codes.
   - ✓ Compiled executable of your application (EXE or APK).
   - ✓ Another supplementary file, if any.

4. **Special Conditions for Project**

   - ✓ All DataGridView column can not be added, edited, or deleted manually**.**
   - ✓ All DataGridView cell value can not be added, edited, or deleted manually**.**

GRAND HOTEL - ENTITY RELATIONSHIP DIAGRAM



GRAND HOTEL - DATA DICTIONARY

Table	PK	FK	Column	Required	Type	Length	Notes
CleaningRoom	PK		ID	Yes	int		Auto Increment/Identity
			Date	Yes	date		
		FK	EmployeeID	Yes	int		
CleaningRoomDetail	PK		ID	Yes	int		Auto Increment/Identity
		FK	CleaningRoomID	Yes	int		
		FK	RoomID	Yes	int		
			StartDateTime	Yes	datetime		
			FinishDateTime	Yes	datetime		
			Note	Yes	text		
			StatusCleaning	Yes	varchar	50	
CleaningRoomItem	PK		ID	Yes	int		Auto Increment/Identity
		FK	CleaningRoomDetailID	Yes	int		
		FK	ItemID	Yes	int		
			Qty	Yes	int		
			Status	Yes	varchar	50	
Customer	PK		ID	Yes	int		Auto Increment/Identity
			Name	Yes	varchar	50	
			NIK	No	varchar	50	
			Email	No	varchar	50	
			Gender	No	char	1	
			PhoneNumber	No	varchar	20	
			Age	No	int		
Employee	PK		ID	Yes	int		Auto Increment/Identity
			Username	Yes	varchar	50	
			Password	Yes	varchar	100	
			Name	Yes	varchar	50	
			Email	Yes	varchar	50	
			Address	Yes	varchar	100	
			DateOfBirth	Yes	date		
		FK	JobID	Yes	int		
Item	PK		ID	Yes	int		Auto Increment/Identity
			Name	Yes	varchar	50	
			RequestPrice	Yes	int		
			CompensationFee	No	int		
ItemStatus	PK		ID	Yes	int		Auto Increment/Identity
			Name	Yes	varchar	50	
Job	PK		ID	Yes	int		Auto Increment/Identity
			Name	Yes	varchar	50	
Reservation	PK		ID	Yes	int		Auto Increment/Identity
			DateTime	Yes	datetime		
		FK	EmployeeID	Yes	int		
		FK	CustomerID	Yes	int		
			Code	Yes	varchar	50	
ReservationCheckOut	PK		ID	Yes	int		Auto Increment/Identity
		FK	ReservationRoomID	Yes	int		
		FK	ItemID	Yes	int		
		FK	ItemStatusID	Yes	int		
			Qty	Yes	int		
			TotalCharge	Yes	int		
ReservationRequestItem	PK		ID	Yes	int		Auto Increment/Identity
		FK	ReservationRoomID	Yes	int		
		FK	ItemID	Yes	int		
			Qty	Yes	int		
			TotalPrice	Yes	int		
ReservationRoom	PK		ID	Yes	int		Auto Increment/Identity
		FK	ReservationID	Yes	int		
		FK	RoomID	Yes	int		
			StartDateTime	Yes	date		
			DurationNights	Yes	int		
			RoomPrice	Yes	int		
			CheckInDateTime	Yes	datetime		
			CheckOutDateTime	Yes	datetime		
Room	PK		ID	Yes	int		Auto Increment/Identity
		FK	RoomTypeID	Yes	int		
			RoomNumber	Yes	varchar	50	
			RoomFloor	Yes	varchar	50	
			Description	No	text		
RoomType	PK		ID	Yes	int		Auto Increment/Identity
			Name	Yes	varchar	50	
			Capacity	Yes	int		
			RoomPrice	Yes	int		

GRAND HOTEL - WIREFRAME

01 Login Form

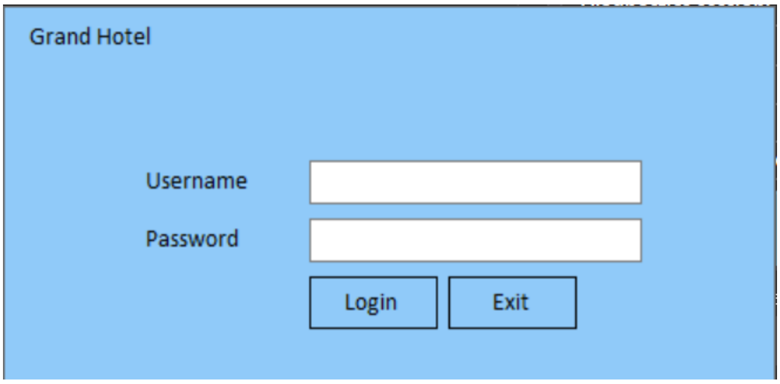


Figure 1. Login Form

02 Main Form



Figure 2. Main Form for Front Office



Figure 3. Main Form for Housekeeper Supervisor



Figure 4. Main Form for Housekeeper



Figure 5. Main Form for Admin

03 Master Room Type

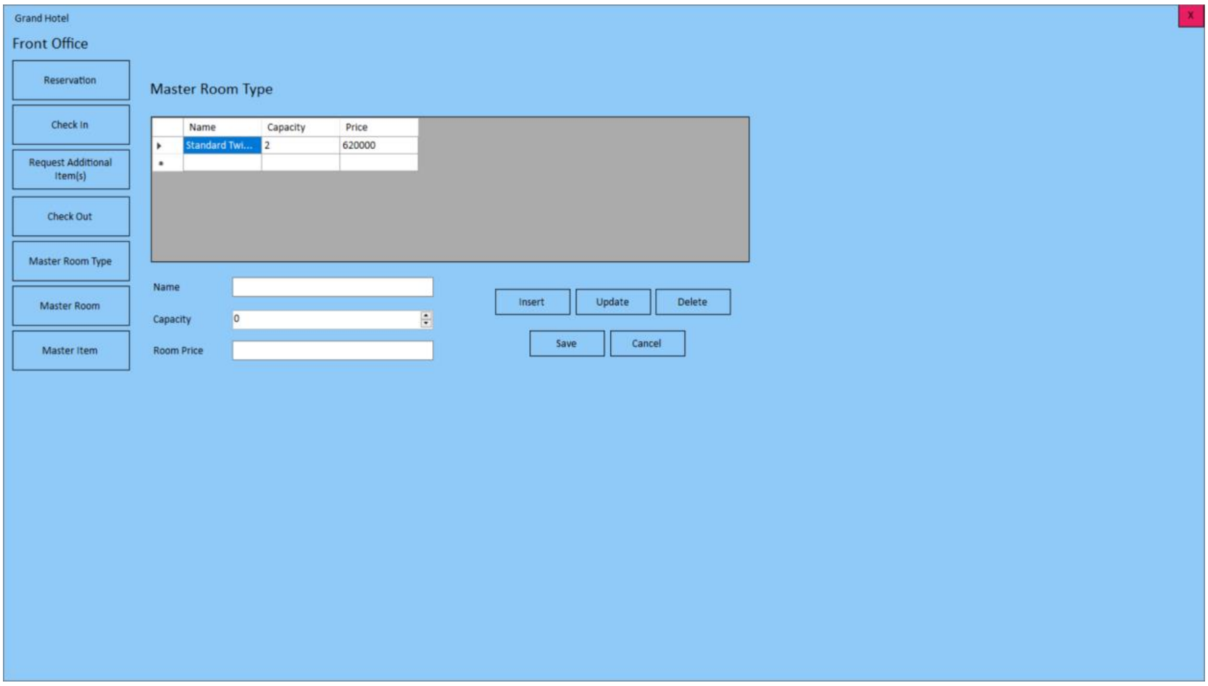


Figure 6. Master Room Type Form

04 Master Room

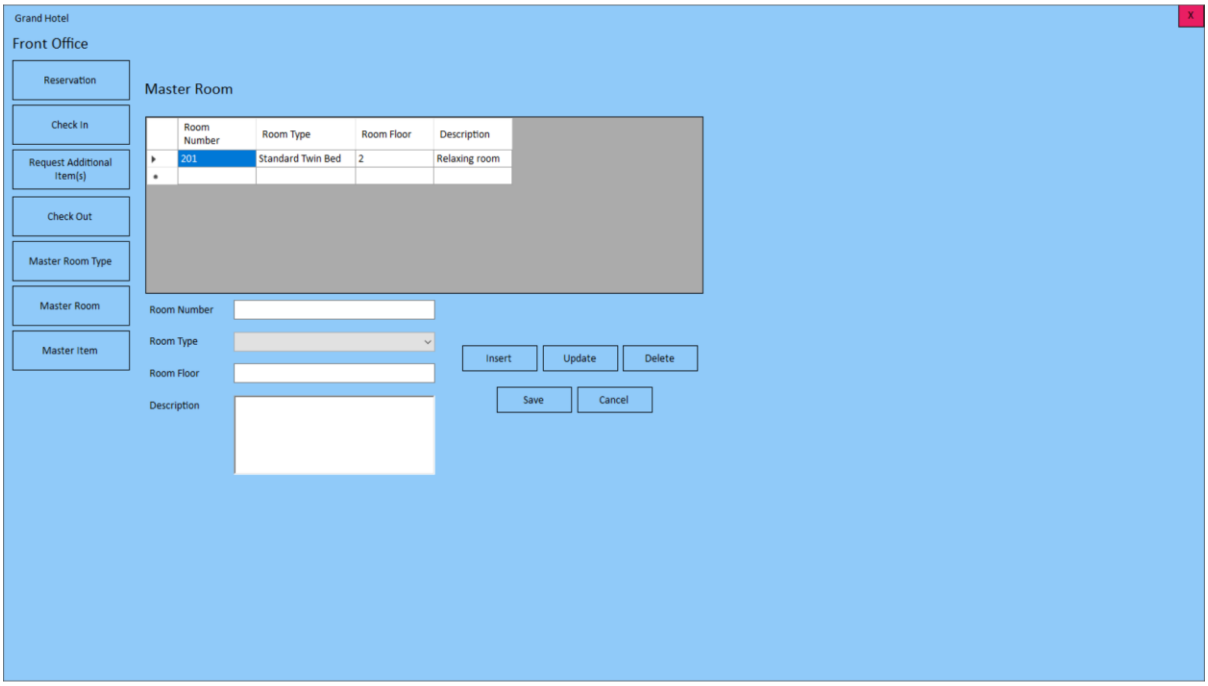


Figure 7. Master Room

05 Master Employee

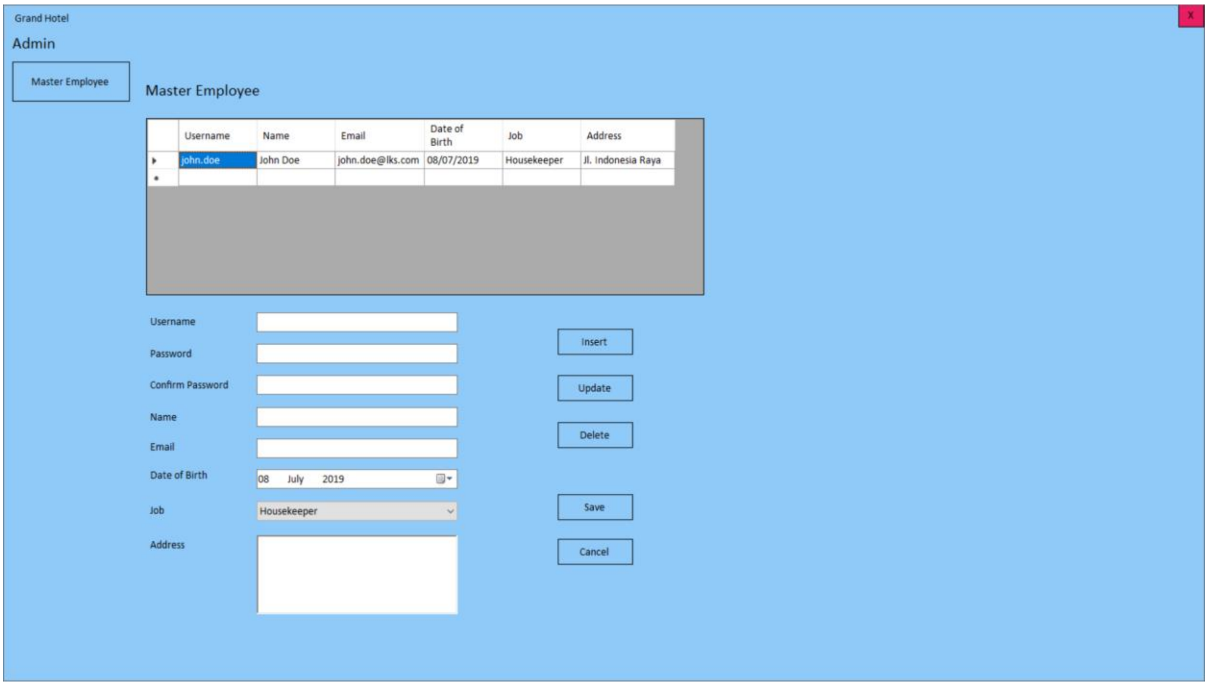


Figure 8. Master Employee Form

06 Master Item

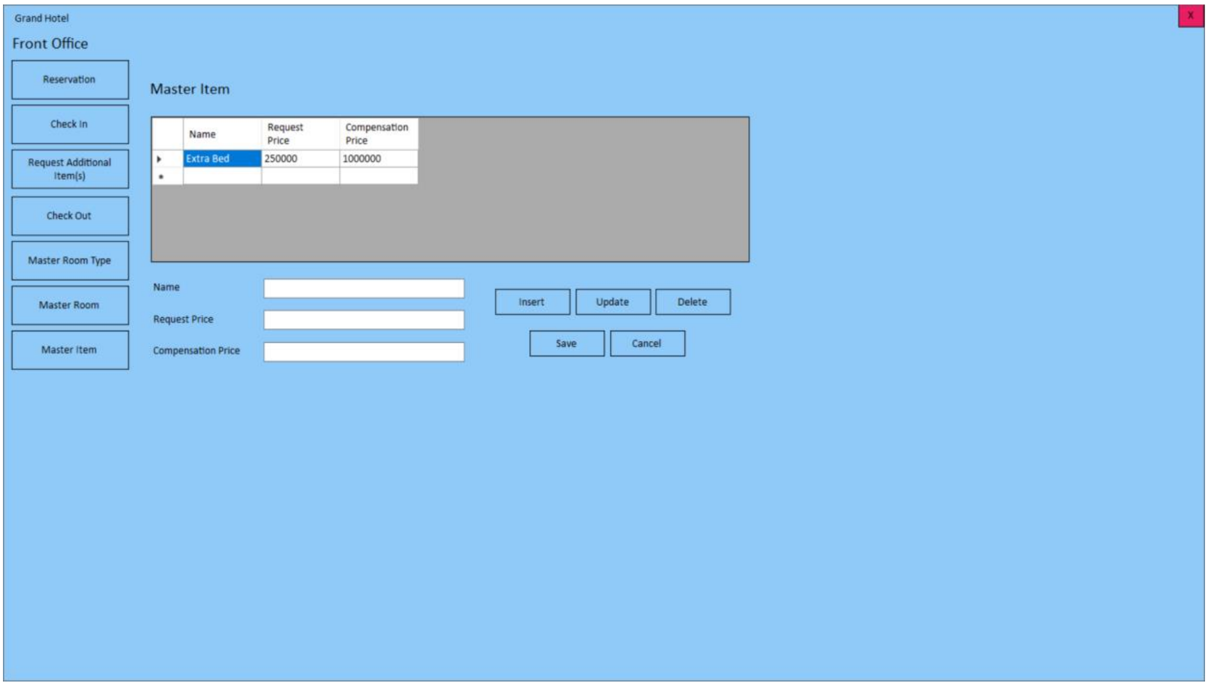


Figure 9. Master Item Form

07 Reservation

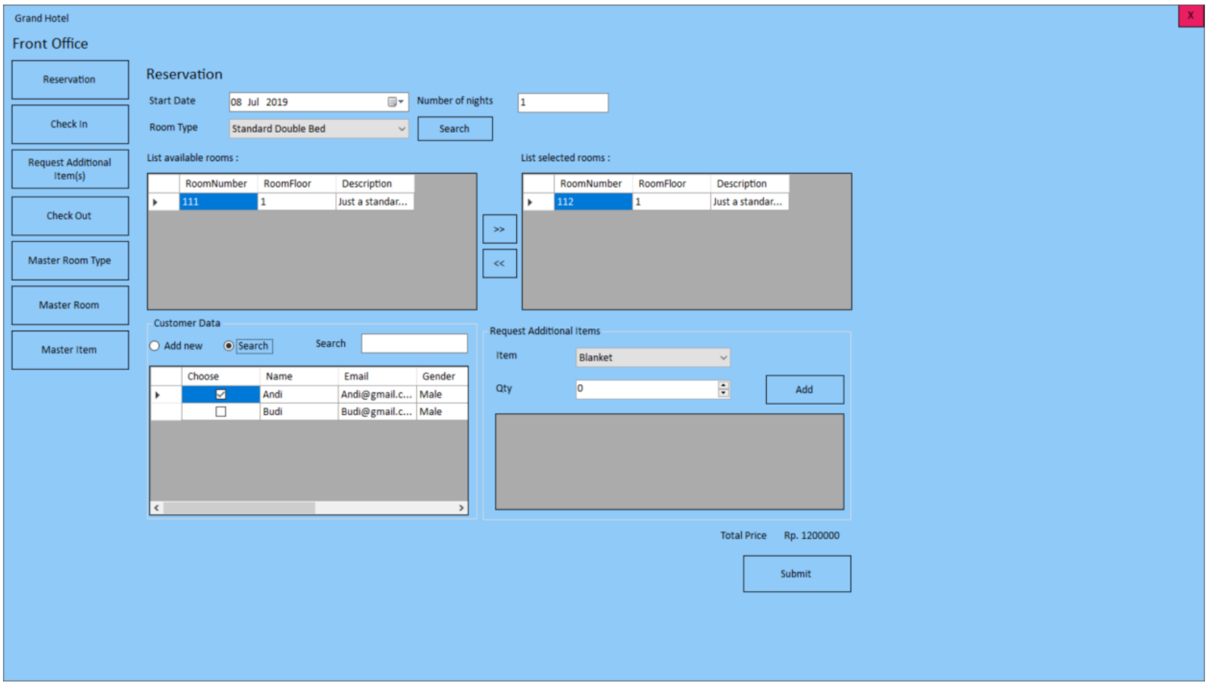


Figure 10. Reservation Form (Search Existing User)

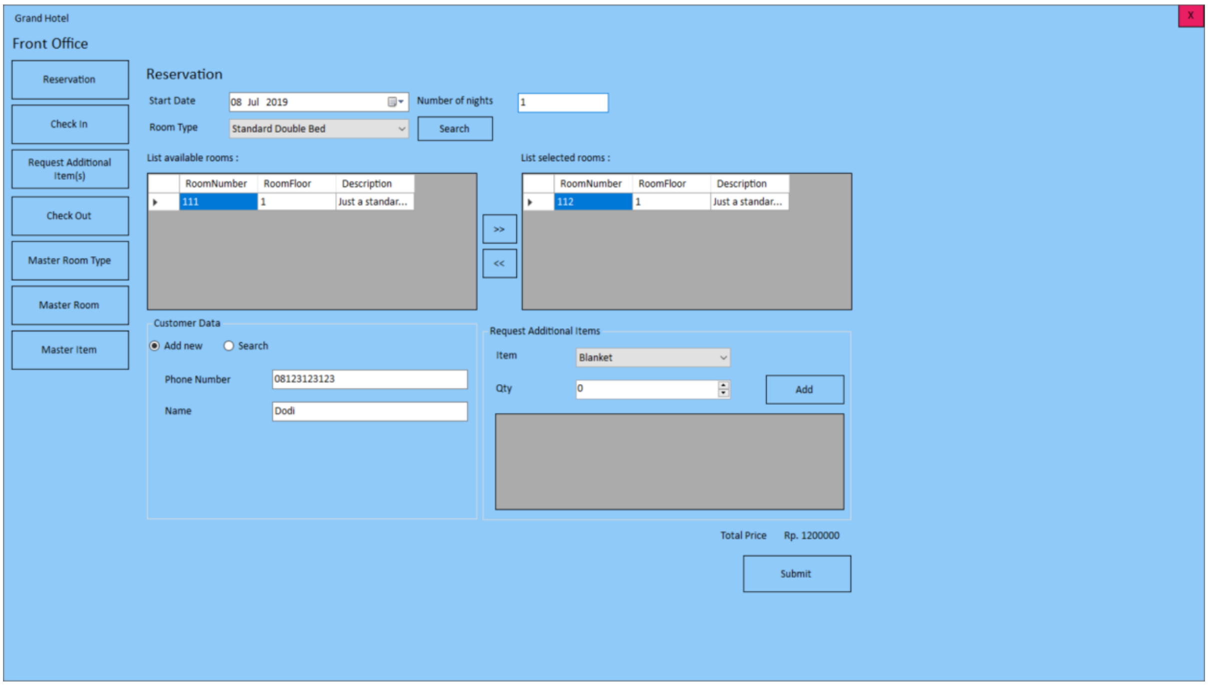


Figure 11. Reservation Form (Add New User)

08 Check In

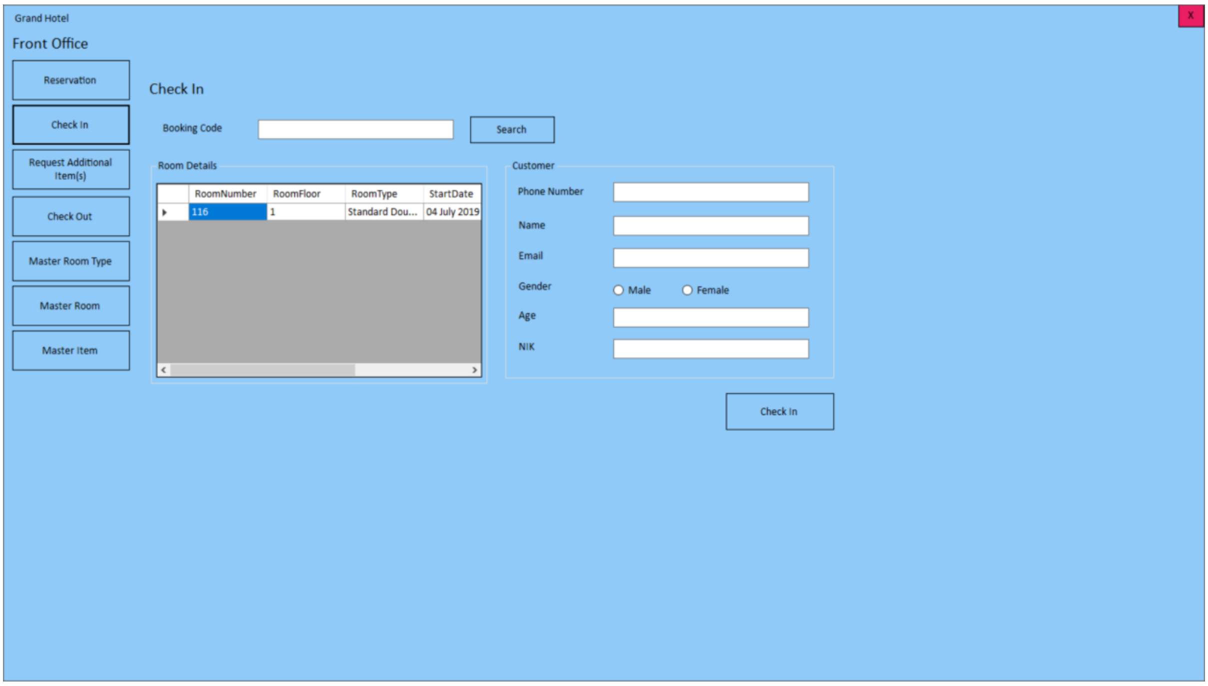


Figure 12. Check In Form

09 Request Additional Item(s)

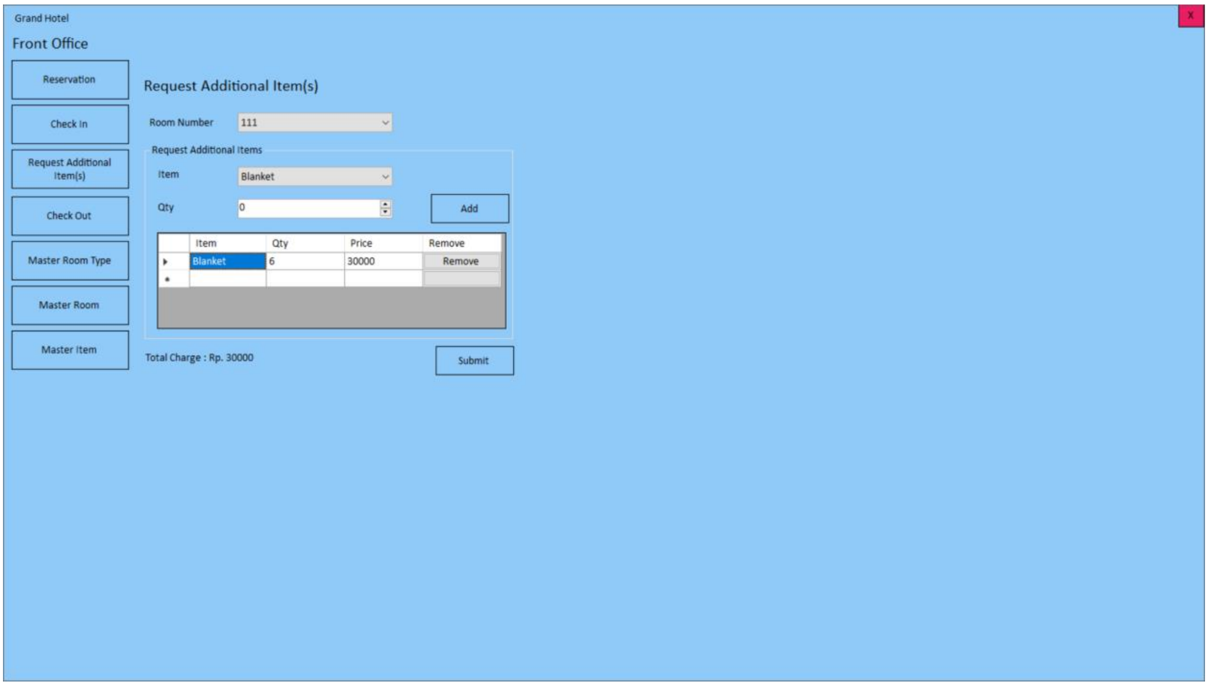


Figure 13. Request Additional Item(s) Form

10 Check Out

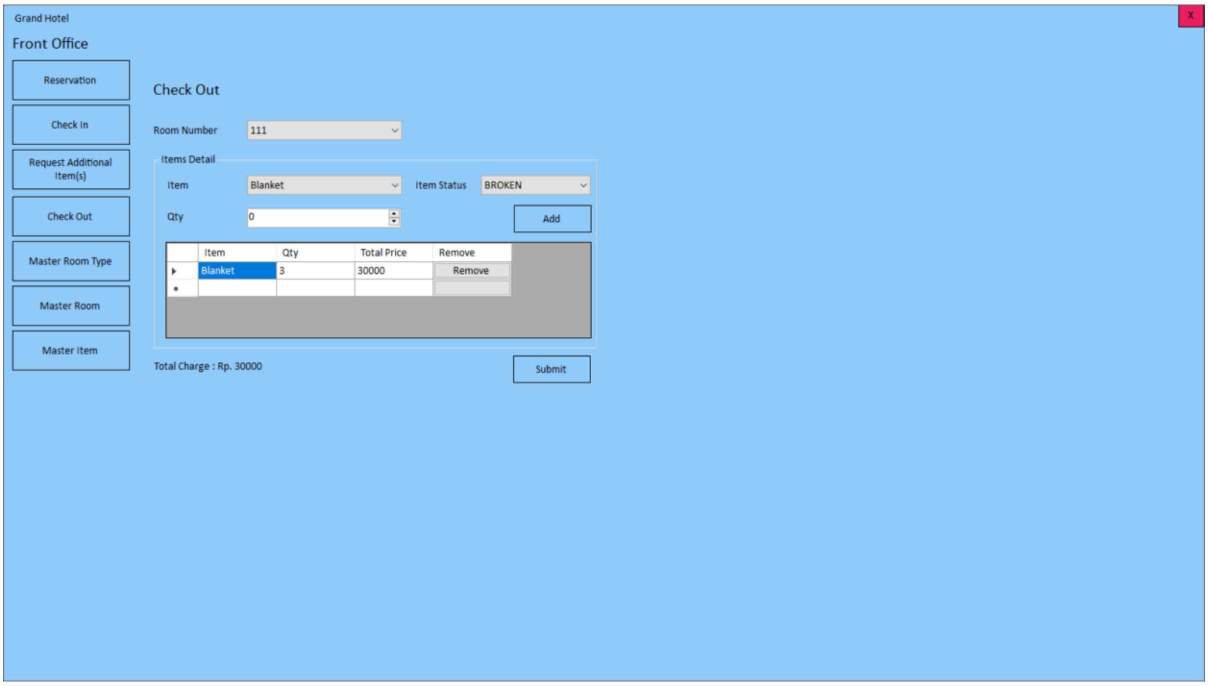


Figure 14. Check Out Form

11 Add Housekeeping Schedule

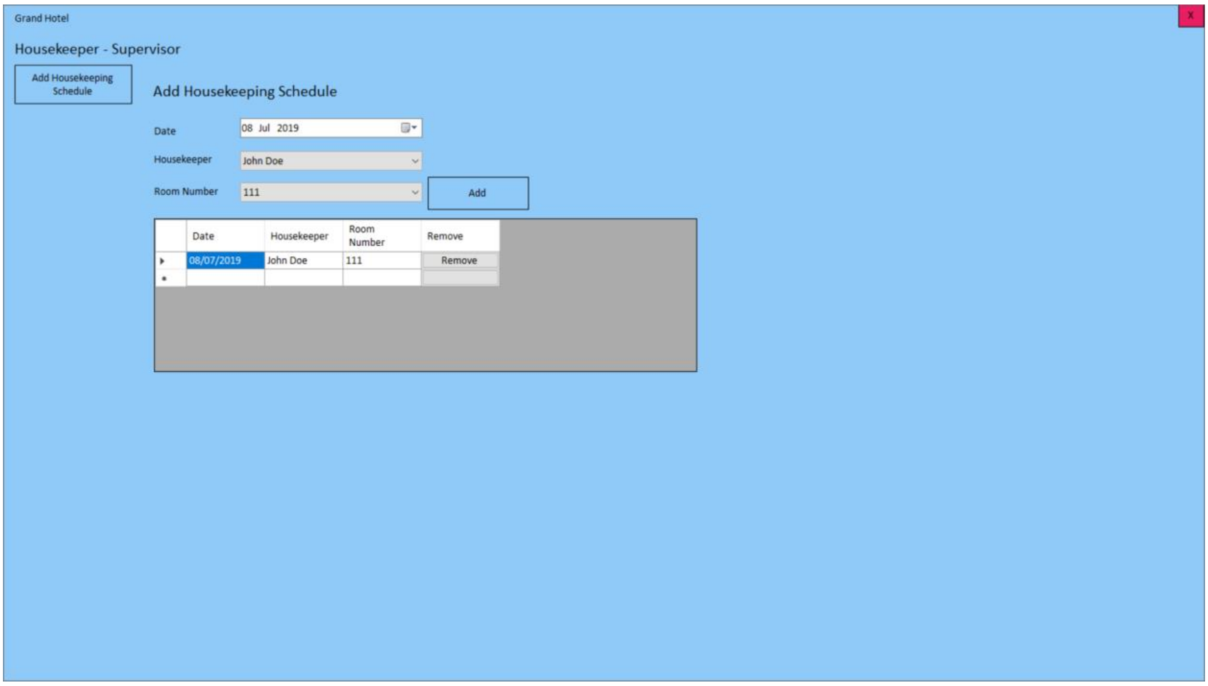


Figure 15. Add Housekeeping Schedule

12 Cleaning Room

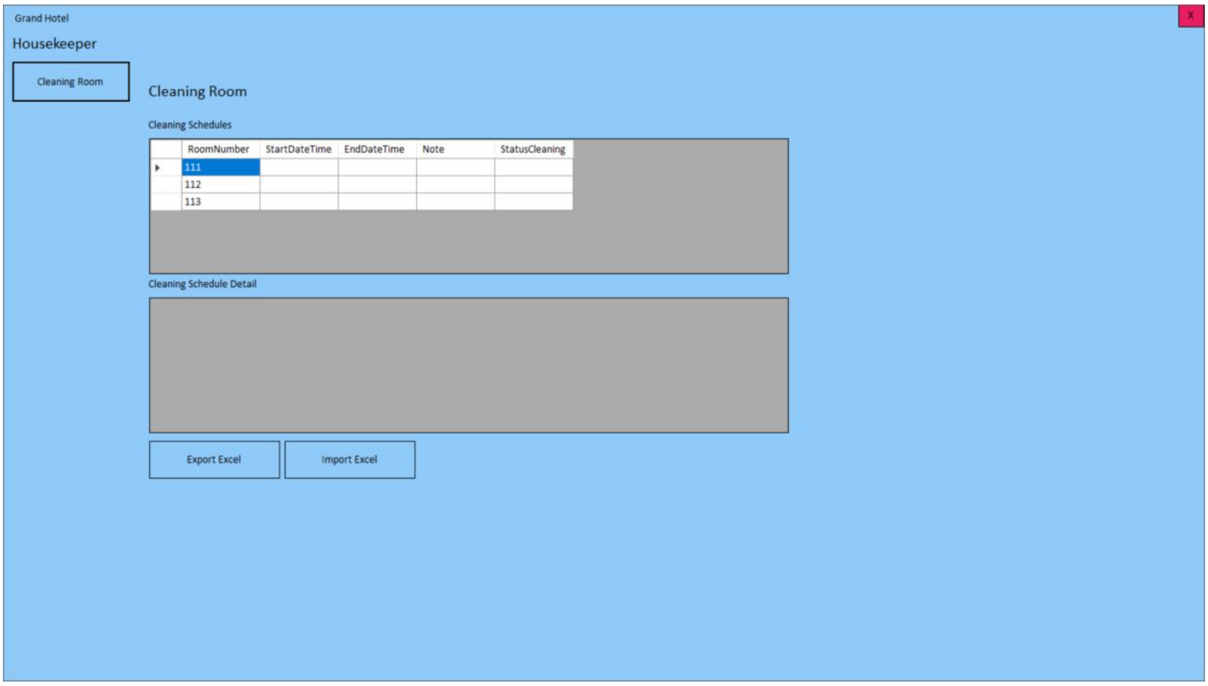


Figure 16. Cleaning Room Form

13 Laundry Hotel

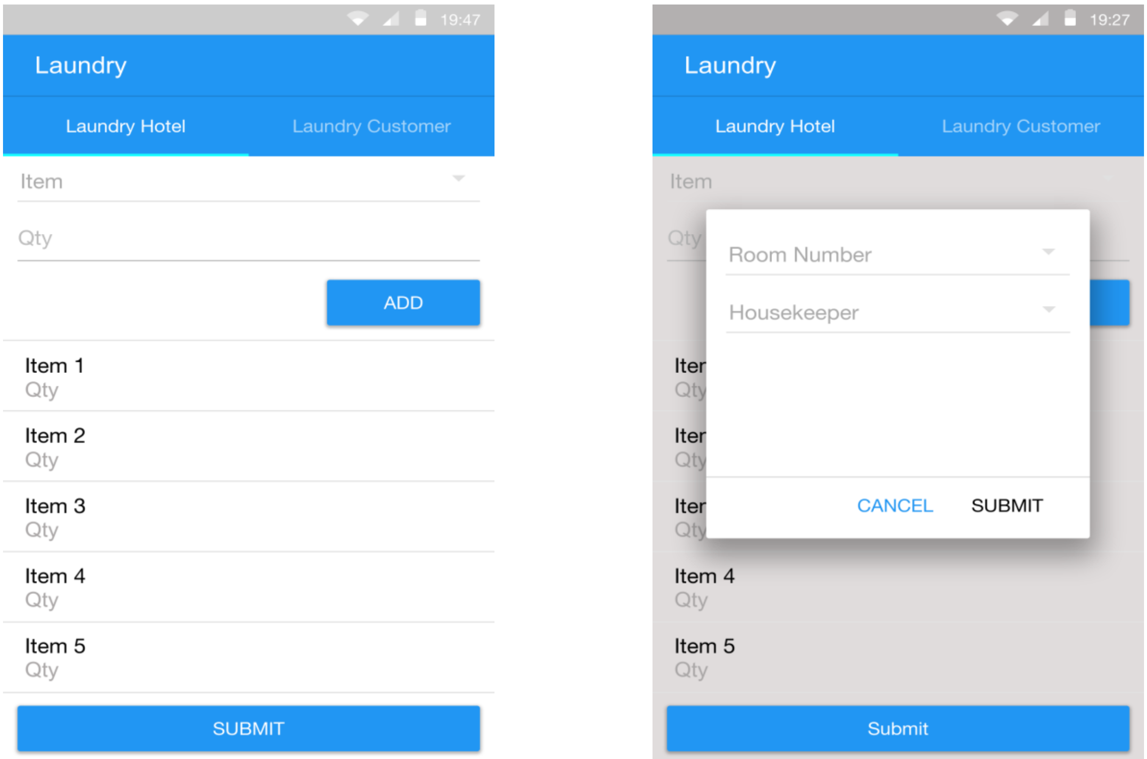


Figure 17. Laundry Hotel Mobile Application Interface

14 Laundry Customer

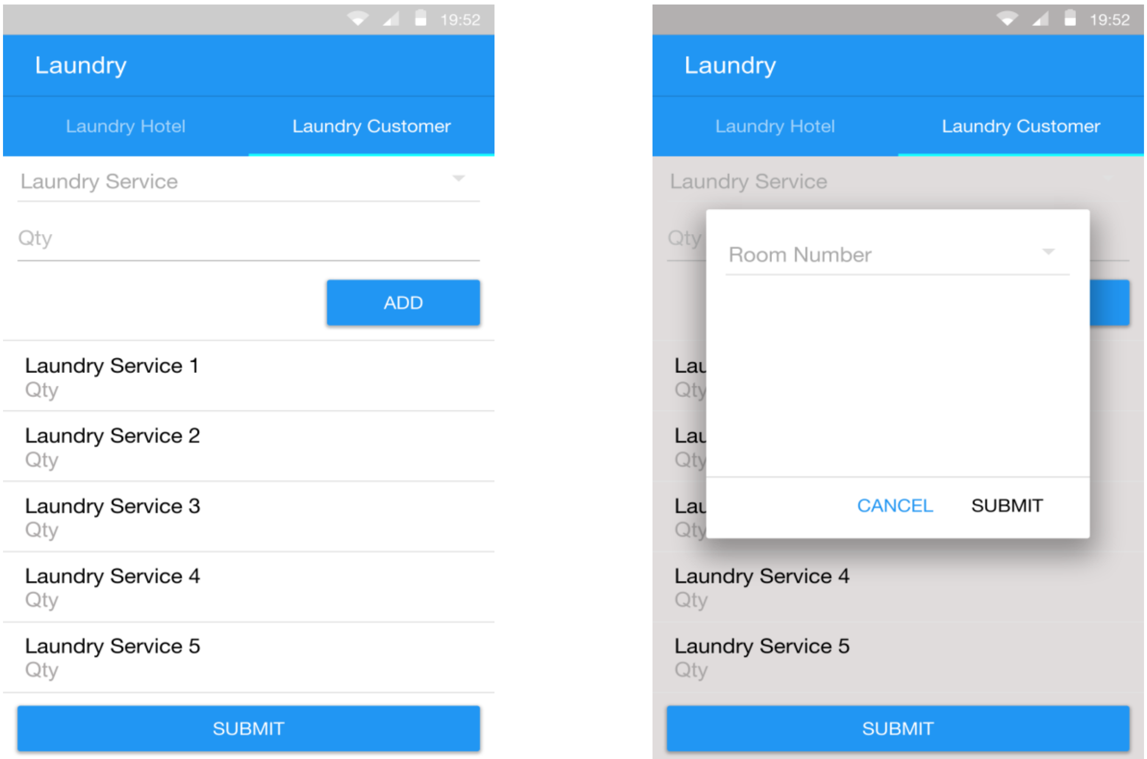


Figure 18. Laundry Customer Mobile Application Interface

THE DEVELOPMENT OF GRAND HOTEL INFORMATION SYSTEM

GRAND Hotel is one of Indonesia famous vacational hotel. As the Hotel growth, the Board of Managements thinks it is time to use Information System to support their business process for transaction process. You have been hired as a famous consultant and programmer to analyze and design the system.

You are required to develop the system based on the given design system. Ensure that your system can accommodate SMK Hotel business process properly, following the instruction on the project!

Module 1: Database Creation and Main Form Development

1. **Database and Table Creation**

   Create a database using Microsoft SQL Server on your local database server named **"PC_XX"** (XX is PC number). After that, create the table required according to the given **ERD** and **Data Dictionary**. Please be aware of the **data type** and the **constraint** of each table given.

2. **Data Insertion**

   Insert the master data given to you from **"data.xlsx"** file into each of the corresponding table on your newly created database.

3. **User Interface Design**

   Design user interface of **Grand Hotel** system information according to the given **wireframe** using your preferred platform, either **C# or Visual Basic.NET**. Your project name should be **"PC_XX"** (XX is PC number). Please be aware for using the right GUI Component as shown on the wireframe.

4. **Develop "01 Login Form"**
   a. Design the form by following **"01 Login Form"** wireframe.
   b. All password stored in the database has been hashed using **SHA256 algorithm.**
   c. Validate that all input must be filled in.
   d. Validate that username and password combination exists in the database.

5. **Develop "02 Main Form"**
   a. Design the main form by following **"02 Main Form"** wireframe.
   b. Menu available will vary according to the user role. Below table shows the role and their available menu.
   c. You are required to develop the Main Form and buttons to access another menu for each user role. **Note that other forms will be created in Module 2**.

Form	Accessible by			
	Admin	Front Office	Housekeeper Supervisor	Housekeeper
Master Room Type		✓		
Master Room		✓		
Master Employee	✓			
Master Item		✓		
Reservation		✓		
Check In		✓		
Request Additional Item(s)		✓		
Check out		✓		
Add Housekeeping Schedule			✓	
Cleaning Room				✓

Module 2: Master Form Development

1. **Master Room Type**
   a. Design the form by following **"03 Master Room Type"** wireframe.
   b. **Initially** the form will have it's:
      - DataGridView filled with all room type data. When a row is clicked, the details of selected row will be displayed in each appropriate component.
      - Button Insert, Update and Delete enabled.
      - Button Save and Cancel disabled.
      - All input component disabled.
   c. When the user **clicks Button Insert**, the form will enter **Insert** mode, then:
      - Button Insert, Update and Delete will be disabled.
      - Button Save and Cancel will be enabled.
      - Enable and clear all input components.
   d. When the user **clicks Button Update**, the form will enter **Update** mode, then:
      - Button Insert, Update and Delete will be disabled.
      - Button Save and Cancel will be enabled.
      - Enable all input components.
   e. When the user **clicks Button Delete**, then:
      - Show delete confirmation.
      - On 'Yes', delete the data and the form return into initial condition.
      - On 'cancel', the process will be cancelled, and the form return into initial condition.
   f. When the user clicks Button **Save**, then:
      - Validate all input value, such as: **empty input, proper data type, proper data formatting, and other crucial validations**.
      - Perform the action according to the form state, either **Insert** or **Update** to the database.
      - Return the form into initial condition.
   g. When the user clicks Button **Cancel**, then:
      - Cancel previously chosen action.
      - Return the form into initial condition.

2. Master Room

a. Design the form by following **“04 Master Room”** wireframe.

b. **Initially** the form will have it’s:
   - DataGridView filled with all room data. When a row is clicked, the details of selected row will be displayed in each appropriate component.
   - DropDown Room Type filled in with room type data.
   - Button Insert, Update and Delete enabled.
   - Button Save and Cancel disabled.
   - All input component disabled.

c. When the user **clicks Button Insert**, the form will enter **Insert** mode, then:
   - Button Insert, Update and Delete will be disabled.
   - Button Save and Cancel will be enabled.
   - Enable and clear all input components.

d. When the user **clicks Button Update**, the form will enter **Update** mode, then:
   - Button Insert, Update and Delete will be disabled.
   - Button Save and Cancel will be enabled.
   - Enable all input components.

e. When the user **clicks Button Delete**, then:
   - Show delete confirmation.
   - On ‘Yes’, delete the data and the form return into initial condition.
   - On ‘cancel’, the process will be cancelled, and the form return into initial condition.

f. When the user clicks Button **Save**, then:
   - Validate all input value, such as: **empty input, proper data type, proper data formatting, and other crucial validations**.
   - Perform the action according to the form state, either **Insert** or **Update** to the database.
   - Return the form into initial condition.

g. When the user clicks Button **Cancel**, then:
   - Cancel previously chosen action.
   - Return the form into initial condition.

3. Master Employee

a. Design the form by following **“05 Master Employee”** wireframe.

b. **Initially** the form will have it’s:
   - DataGridView filled with employee data. When a row is clicked, the details of selected row will be displayed in each appropriate component.
   - Button Insert, Update and Delete enabled.
   - Button Save and Cancel disabled.
   - All input component disabled.

c. When the user **clicks Button Insert**, the form will enter **Insert** mode, then:
   - Button Insert, Update and Delete will be disabled.
   - Button Save and Cancel will be enabled.
   - Enable and clear all input components.

d. When the user **clicks Button Update**, the form will enter **Update** mode, then:
   - Button Insert, Update and Delete will be disabled.
   - Button Save and Cancel will be enabled.
   - Enable all input components.

e. When the user **clicks Button Delete**, then:
   - Show delete confirmation.
   - On ‘Yes’, delete the data and the form return into initial condition.
   - On ‘cancel’, the process will be cancelled, and the form return into initial condition.

f. When the user clicks Button **Save**, then:
   - Validate all input value, such as: **empty input, proper data type, proper data formatting, and other crucial validations**.
   - Perform the action according to the form state, either **Insert** or **Update** to the database.
   - Return the form into initial condition.

g. When the user clicks Button **Cancel**, then:
   - Cancel previously chosen action.
   - Return the form into initial condition.

4. Master Item

a. Design the form by following **“05 Master Item”** wireframe.

b. **Initially** the form will have it’s:
   - DataGridView filled with item data. When a row is clicked, the details of selected row will be displayed in each appropriate component.
   - DropDown Job filled in with job data.
   - Button Insert, Update and Delete enabled.
   - Button Save and Cancel disabled.
   - All input component disabled.

c. When the user **clicks Button Insert**, the form will enter **Insert** mode, then:
   - Button Insert, Update and Delete will be disabled.
   - Button Save and Cancel will be enabled.
   - Enable and clear all input components.

d. When the user **clicks Button Update**, the form will enter **Update** mode, then:
   - Button Insert, Update and Delete will be disabled.
   - Button Save and Cancel will be enabled.
   - Enable all input components.

e. When the user **clicks Button Delete**, then:
   - Show delete confirmation.
   - On ‘Yes’, delete the data and the form return into initial condition.
   - On ‘cancel’, the process will be cancelled, and the form return into initial condition.

f. When the user clicks Button **Save**, then:
   - Validate all input value, such as: **empty input, proper data type, proper data formatting, and other crucial validations**.
   - Perform the action according to the form state, either **Insert** or **Update** to the database.
   - Return the form into initial condition.

g. When the user clicks Button **Cancel**, then:
   - Cancel previously chosen action.
   - Return the form into initial condition.

5. Add Housekeeping Schedule

a. Design the form by following **"11 Add Housekeeping Schedule"** wireframe.

b. **Initially**, the condition of the form is:

- DataGridView is empty.
- **DropDown Housekeeper** is filled in with all Housekeeper name.
- **DropDown Room Number** is filled in with all room number.
- Button Add is enabled.

c. DataGridView will show cleaning schedule of the selected housekeeper for the selected date. The DataGridView content will be reloaded everytime DropDown Housekeeper or DateTimePicker is changed.

d. When the user **clicks Button Add**, a new cleaning schedule will be added to the housekeeper data for the selected date.

e. When the user clicks **Button Remove**, delete the selected cleaning schedule.

Module 3: Transaction Form Development

1. Reservation

a. Design the form by following **“07 Reservation”** wireframe.

b. **Initially,** the condition of the form is:
   - DropDown Room Type is filled in with room type data.
   - RadioButton Search is selected.
   - DropDown Item is filled in with all items.

c. **DataGridView Available Rooms** shows available room whose type matched with DropDown Room Type. The DataGridView will be reloaded everytime DropDown Room Type is changed. Make sure that available room is validated with all other reservation data on the selected period.

d. When the user clicks **Button >>**, the selected available room will be added to the booking list.

e. When the user clicks **Button <<**, the selected room will be removed from booking list then added into the available room list.

f. When RadioButton Search is active, TextBox Search can be used to search existing user by name and display the result into DataGridView Customer Data. The whole rooms booked, then will be booked under the checked user. Only one user can be checked from the DataGridView Customer Data.

g. When the user clicks **Button Add** then**:**
   - The data will be shown in the DataGridView Additional Item.
   - Same item cannot be added twice, instead it will update the quantity.

h. User can remove additional item from DataGridView Additional Item by clicking remove button in each row.

i. Update Total Price label each time there is any change in the Total Price. The Total price is calculated as follow:

$$TotalPrice = \left(\sum_{1}^{n} roomPrice_n \times numberOfNight_n\right) + \left(\sum_{0}^{m} itemPrice_m \times quantity_m)\right)$$

   Where n is number of booked room and m is number of added additional items.

j. When the user clicks **Button Submit** then:
   - Validate that all input required has been filled
   - Insert the reservation data into the database
   - Display success message and booking code. The booking code should be a unique uppercase alphanumeric of six characters.

2. **Check In**
   a. Design the form by following **“08 Check In”** wireframe.
   b. When the user clicks **Button Search,** then display all rooms that booked under the input search into DataGridView Room Details. Display only rooms that has not been checked in yet. If all rooms have been checked in, show an error message. If the input booking code is not found, then show an error message.
   c. When the user type in TextBox Phone Number, check with the existing user data. If a match is found, then:
      - Auto fill the remaining input field.
      - Any changes made into the field, will be updated into the customer data during the check in process.

      If no phone number match is found, then the user must fill in the remaining field and this data will be counted as a new customer.
   d. When the user clicks **Button Check In:**
      - Save the check in data and update or insert the customer data into the database.
      - Reload DataGridView Room Details.

3. **Request Additional Item(s)**
   a. Design the form by following **“09 Request Additional Item(s)”** wireframe.
   b. **DropDown Room Number** is filled in with all room number currently under reservation or has guest staying in.
   c. DropDown Item is filled in with all items data.
   d. When the user clicks **Button Add** then**:**
      - The data will be shown in the DataGridView Additional Item.
      - Same item cannot be added twice, instead it will update the quantity.
   e. User can remove additional item from DataGridView Additional Item by clicking remove button in each row.
   f. Update Total Price label each time there is any change in the Total Price. The Total price is calculated as follow:

$$TotalPrice = \left(\sum_{0}^{m} itemPrice_m \times quantity_m) \right)$$

      Where m is number of added additional items.
   g. When the user clicks **Button Submit** then:
      - Validate that all input required has been filled
      - Insert the additional item(s) data into the database

4. Check Out

a. Design the form by following **“10 Check Out”** wireframe.

b. **DropDown Room Number** is filled in with all room number has guest staying in.

c. DropDown Item shows all items data.

d. DropDown Item Status is filled in with all available item status.

e. When the user clicks **Button Add** then**:**

- The data will be shown in the DataGridView Item.
- Same item cannot be added twice, instead it will update the quantity.

f. User can remove item from DataGridView Item by clicking remove button in each row.

g. Update Total Price label each time there is any change in the Total Price. The Total price is calculated as follow:

$$TotalPrice = \left(\sum_{0}^{m} itemPrice_m \times quantity_m) \right)$$

Where m is number of added items.

h. When the user clicks **Button Submit,** save the check out data into the database.

Module 4: Transaction Form Development and Database Analysis

1. Cleaning Room

a. Design the form by following **"12 Cleaning Room"** wireframe.

b. **Initially,** the condition of the form is:

- DataGridView Cleaning Schedules shows all cleaning schedules assigned to the user for that day.
- Clicking on one of the schedules will shows the detail in the DataGridView Cleaning Schedules Detail.
- DataGridView Cleaning Schedules Detail shows all items condition for the selected cleaning schedule.

c. When the user clicks **Button Export Excel** then:

- An excel (*.xlsx) file will be generated as follow:

	A	B	C	D	E	F	G	H
1	**Date**	04 Jul 2019						
2	**Employee**	John Doe						
3								
4	**RoomNumber**	**StartDateTime**	**EndDateTime**	**Note**	**StatusCleaning**	**Item**	**Qty**	**Status**
5	111							
6	112							
7	113							

Figure 17. Excel Format for Export Excel Feature in Cleaning Room Form

- Make sure to fill in the date and employee name.
- The excel file name must be: **"EmployeeName_DDMMYYYY.xlsx"**

d. When the user clicks **Button Import Excel** then:

- User can upload the previously downloaded file. Validate the file type and file name.
- Detail of all items condition will be stored in the database
- Reload DataGridView Cleaning Schedules Detail.
- If in the future the user uploads the file again, then previously uploaded data will be overwritten.
- Below is the example of the uploaded excel file:

	A	B	C	D	E	F	G	H
1	**Date**	04 Jul 2019						
2	**Employee**	John Doe						
3								
4	**RoomNumber**	**StartDateTime**	**EndDateTime**	**Note**	**StatusCleaning**	**Item**	**Qty**	**Status**
5	111	04 July 2019 09:00	04 July 2019 09:30		CLEANED	blanket	1	TIDY
6						bath soap	2	CHANGE
7	112	04 July 2019 09:30	04 July 2019 10:00		CLEANED	blanket	1	TIDY
8	113	04 July 2019 10:00	04 July 2019 10:30		SKIP			

Figure 18. Sample of Uploaded File in Cleaning Room Form

- Below is sample of the Cleaning Room Form before and after import data:

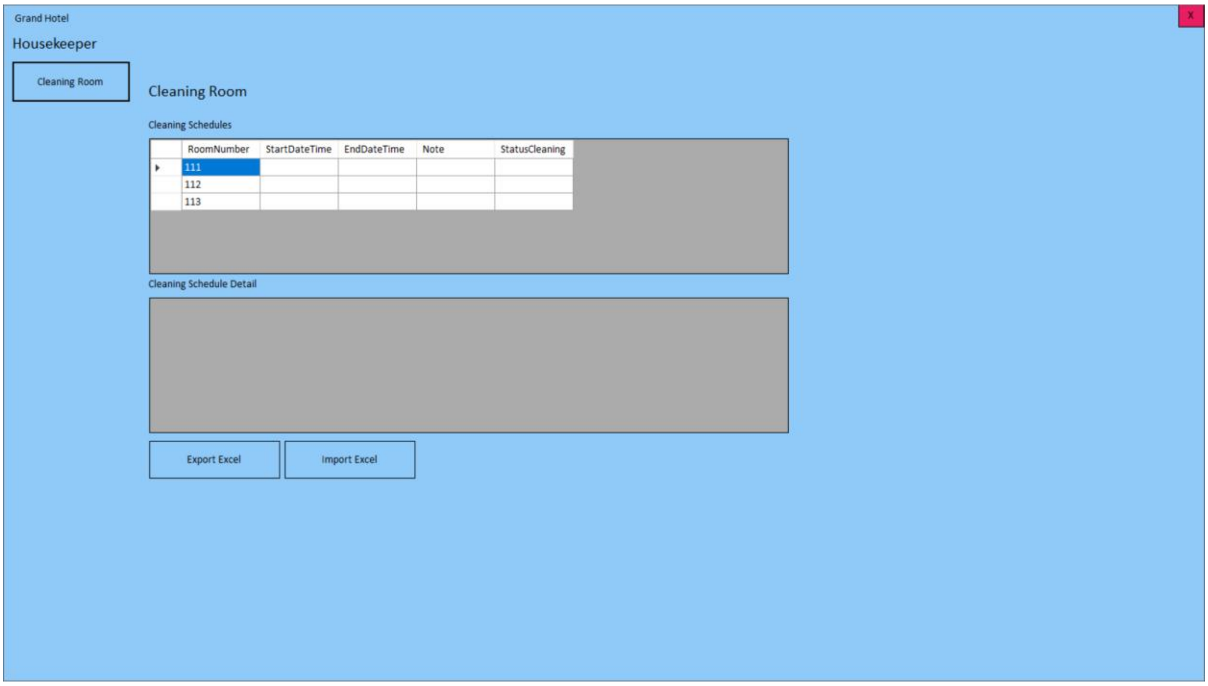


Figure 19. Cleaning Room Form (Before Import Data)

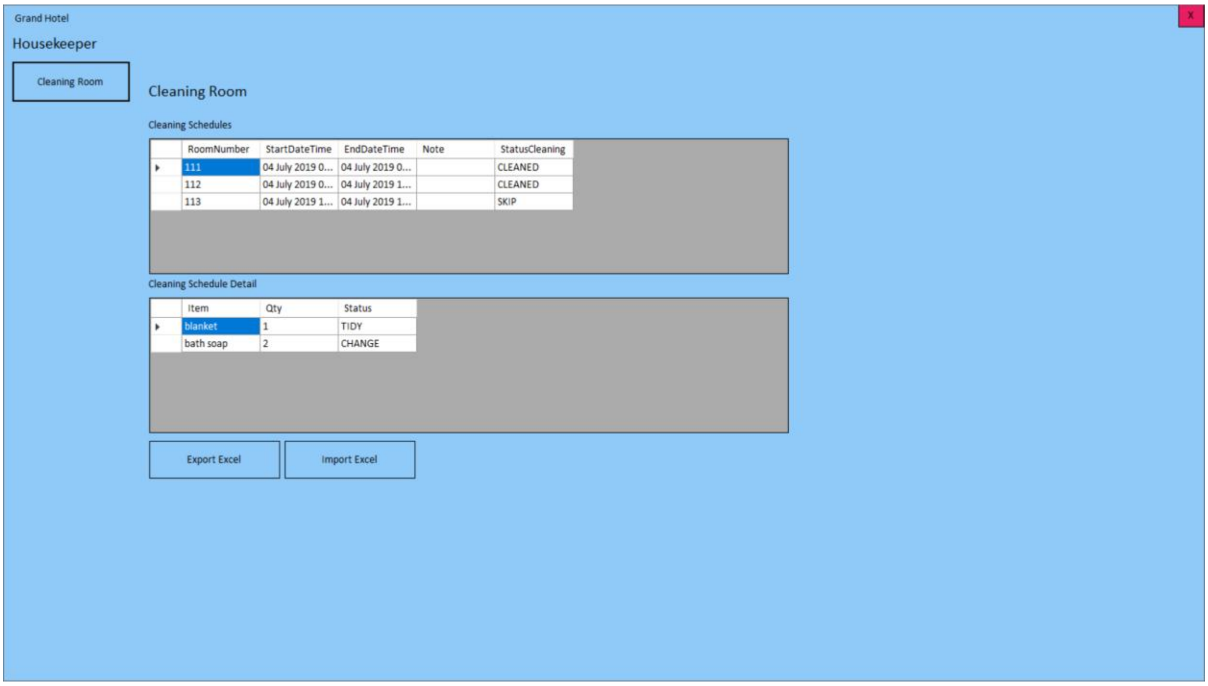


Figure 20. Cleaning Room Form (After Import Data)

2. Database Analysis

GRAND Hotel Laundry provides laundry services for their internal inventory items and customers items. The laundry capable of cleaning many types of clothing and fabric, such as: linen blanket, cotton shirt, flannel skirt, etc. The rate is determined by the types of the items.

Customers may also choose different duration of services, such as: normal (2 days), over night service (1 night), and express (2 hours). The price increases along with the duration of service. Customer can also leave a note or special request to the laundry staff. Manager and receptionist can monitor all the laundry progress. The laundry staff will update the laundry progress in case the guest asking for confirmation.

When handing over the laundry items to the customer, the laundry staff name will be recorded. Customer can choose to pay the services at that time or ask the staff to include it to their room charge.

As the system analyst, you are asked to create:

a. Database design in the form of Entity Relationship Diagram (ERD). Make sure that your ERD provide relationship and multiplications between tables. Your submission should be named **"PC_XX. vsdx"** (XX is PC number).
b. Data dictionary in excel. Your submission should be named **"PC_XX.xlsx"** (XX is PC number).
c. SQL script to create the tables and sample data (minimum 5 for each table). Your submission should be named **"PC_XX.sql"** (XX is PC number).
d. Wireframe of the proposed new system. Your submission project name should be named **"PC_XX_Modul3.2"** (XX is PC number).

Module 5: Mobile Application Development

1. API Documentation

This module is no longer use database created in the previous module. Below is the API documentation for your test project this module.

a. Items

Request	
Method	**GET**
URL	[IP Server]/api/items

b. Services

Request	
Method	**GET**
URL	[IP Server]/api/services

c. Housekeepers

Request	
Method	**GET**
URL	[IP Server]/api/employees

d. Rooms

Request	
Method	**GET**
URL	[IP Server]/api/rooms

e. Laundry Hotel

<table>
<tr><th colspan="3">Request</th></tr>
<tr><td>Method</td><td colspan="2">POST</td></tr>
<tr><td>URL</td><td colspan="2">[IP Server]/api/laundry/hotel</td></tr>
<tr><td>Format Request Body</td><td colspan="2">{
“RoomID”: <Room ID>,
“EmployeeID”: <Employee ID>,
“Items”: [
{
“ItemID”: <Item ID>,
“Qty”: <Quantity>
}
, …
]
}</td></tr>
<tr><td rowspan="3">Response</td><td>HTTP Code</td><td>Status</td></tr>
<tr><td>200</td><td>Error</td></tr>
<tr><td>422</td><td>Fail (missing field / unexpected value)</td></tr>
</table>

f. Laundry Customer

<table>
<tr><th colspan="3">Request</th></tr>
<tr><td>Method</td><td colspan="2">POST</td></tr>
<tr><td>URL</td><td colspan="2">[IP Server]/api/laundry/customer</td></tr>
<tr><td>Format Request Body</td><td colspan="2">{
“RoomID”: <Room ID>,
“Services”: [
{
“ServiceID”: <Service ID>,
“Qty”: <Quantity>,
“Price”: <Total Price>
}
, …
]
}</td></tr>
<tr><td rowspan="3">Response</td><td>HTTP Code</td><td>Status</td></tr>
<tr><td>200</td><td>Error</td></tr>
<tr><td>422</td><td>Fail (missing field / unexpected value)</td></tr>
</table>

3. Laundry Hotel

a. Design the form by following **“13 Laundry Hotel”** wireframe.

b. DropDown Item will be filled in with items data

c. When the user clicks **Button Add** then:

- Validate Qty field
- Show the added data below the form

d. When the user clicks **Button Submit** then:
    - Show confirmation dialog
    - DropDown Room Number will be filled in with room numbers data.
    - DropDown Housekeeper will be filled in with housekeepers data.
    - On cancel, cancel the process.
    - On submit, insert the data into database via API, show success message, then clear the form.

2. Laundry Customer

a. Design the form by following **“14 Laundry Customer”** wireframe.
b. DropDown Laundry Service will be filled in with service data
c. When the user clicks **Button Add** then:
    - Validate Qty field
    - Show the added data below the form
d. When the user clicks **Button Submit** then:
    - Show confirmation dialog
    - DropDown Room Number will be filled in with room numbers data.
    - On cancel, cancel the process.
    - On submit, insert the data into database via API, show success message, then clear the form.

- Good luck and do your best! -